GSM based Distribution Transformer Monitoring System

Ansuman Sharma (109EE0305)

Rajesh Behura (109EE0260)

Department of Electrical Engineering
National Institute of Technology Rourkela

GSM based Distribution Transformer Monitoring System

A THESIS IN PARTIAL FULFILMENTS OF REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

Bachelor of Technology
in

Electrical Engineering

BY

Ansuman Sharma (109EE0305)

Rajesh Behura (109EE0260)

Under guidance of

Prof. Susmita Das

Department of Electrical Engineering

National Institute of Technology

Rourkela-769008 (ODISHA)

May-2013

DEPARTMENT OF ELECTRICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA- 769 008
ODISHA, INDIA

CERTIFICATE

This is to certify that the draft report/thesis titled "GSM based distribution transformer monitoring system", submitted to the National Institute of Technology, Rourkela by **Mr. Ansuman Sharma, Roll No**: **109EE0305 and Mr. Rajesh Behura, Roll No: 109EE0260** for the award of Bachelor of Technology in Electrical Engineering, is a bonafide record of research work carried out by him under my supervision and guidance.

The candidate has fulfilled all the prescribed requirements.

The draft report/thesis which is based on candidate's own work, has not submitted elsewhere for a degree/diploma.

In my opinion, the draft report/thesis is of standard required for the award of a Bachelor of Technology in Electrical Engineering.

Prof. Susmita Das

Supervisor

Department of Electrical Engineering

National Institute of Technology

Rourkela – 769 008 (ODISHA)

ACKNOWLEDGEMENTS

I am extremely grateful to The Department of Electrical Engineering, for giving us the opportunity to carry out this project, which is an integral part of the curriculum in B. Tech program at the National Institute of Technology, Rourkela.

I would like to express my earnest gratitude and regards to my project guide, Professor Susmita Das, Department of Electrical Engineering, for being the corner stone of my project. It was her perpetual motivation and guidance during the period of doubts and uncertainties that has helped me to carry on with this project.

Ansuman Sharma

Rajesh Behura

Abstract

This project is about design and implementation of a mobile embedded system to monitor and record key parameters of a distribution transformer like load currents, oil level and ambient temperature. The idea of on-line monitoring system integrates a global service mobile (GSM) Modem, with a standalone single chip microcontroller and different sensors. It is installed at the distribution transformer site and the above parameters are recorded using the analog to digital converter (ADC) of the embedded system. The obtained parameters are processed and recorded in the system memory. If any abnormality or an emergency situation occurs the system sends SMS (short message service) messages to the mobile phones containing information about the abnormality according to some predefined instructions programmed in the microcontroller. This mobile system will help the transformers to operate smoothly and identify problems before any catastrophic failure.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In power systems, distribution transformer is electrical equipment which distributes power to the low-voltage users directly, and its operation condition is an important component of the entire distribution network operation. Operation of distribution transformer under rated condition(as per specification in their nameplate) guarantees their long life .However, their life is significantly reduced if they are subjected to overloading, resulting in unexpected failures and loss of supply to a large number of customers thus effecting system reliability. Overloading and ineffective cooling of transformers are the major causes of failure in distribution transformers[2]-[4]. The monitoring devices or systems which are presently used for monitoring distribution transformer exist some problems and deficiencies. Few of them are mentioned below.

(1) Ordinary transformer measurement system generally detects a single transformer parameter, such as power, current, voltage, and phase. While some ways could detect multi-parameter, the time of acquisition and operation parameters is too long, and testing speed is not fast enough.

(2) Detection system itself is not reliable. The main performance is the device itself instability, poor anti-jamming capability, low measurement accuracy of the data, or even state monitoring system should is no effect.

(3) Timely detection data will not be sent to monitoring centers in time, which can not judge distribution transformers three-phase equilibrium.

(4) A monitoring system can only monitor the operation state or guard against steal the power, and is not able to monitor all useful data of distribution transformers to reduce costs.

(5) Many monitoring systems use power carrier communication to send data, but the power carrier communication has some disadvantages: serious frequency interference, with the increase in distance the signal attenuation serious, load changes brought about large electrical

noise. So if use power carrier communication to send data, the real-time data transmission, reliability cannot be guaranteed.

According to the above requirements, we need a distribution transformer real-time monitoring system to detect all operating parameters operation, and send to the monitoring centre in time. It leads to online monitoring of key operational parameters of distribution transformers which can provide useful information about the health of transformers which will help the utilities to optimally use their transformers and keep the asset in operation for a longer period. This will help to identify problems before any serious failure which leads to a significant cost savings and greater reliability. Widespread use of mobile networks and GSM devices such GSM modems and their decreasing costs have made them an attractive option not only for voice media but for other wide area network applications.

CHAPTER 2

BACKGROUND OF THE PROJECT

Abnormality in distribution transformer is accompanied with variation in different parameters like Winding temperature , Top and bottom oil temperatures, Ambient temperature, Load current, Oil flow (pump motor), Moisture in oil ,Dissolved gas in oil, Bushing condition, LTC monitoring, Oil level. However, we are dealing with oil temperature and load current.

Online monitoring system consists of embedded system, GSM modem, mobile-users and GSM networks and sensors installed at transformer site Sensors are installed on transformer side which reads and measures the physical quantity from the distribution transformer and then it converts it into the analog signal. The embedded module is located at the transformer site. It is utilized to acquire, process, display, transmit and receive the parameters to/ from the GSM modem. The second is the GSM module. It is the link between the embedded system and the public GSM network. The third is utility module that has a PC-based -server located at the utility control center. The server is attached to GSM modem and received transmits SMS from/to the transformer site via the GSM module.

CHAPTER 3

HARDWARE IMPLEMENTATION

This chapter explains regarding the Hardware Implementation of the project. It tells about the design and working of the design with the help of circuit diagram and explanation of circuit diagram in detail. It explains the features, programming and serial communication of ATmega 16L microcontroller. It also explains the different modules used in this project.

3.1 INTERFACING MODULE SCHEME

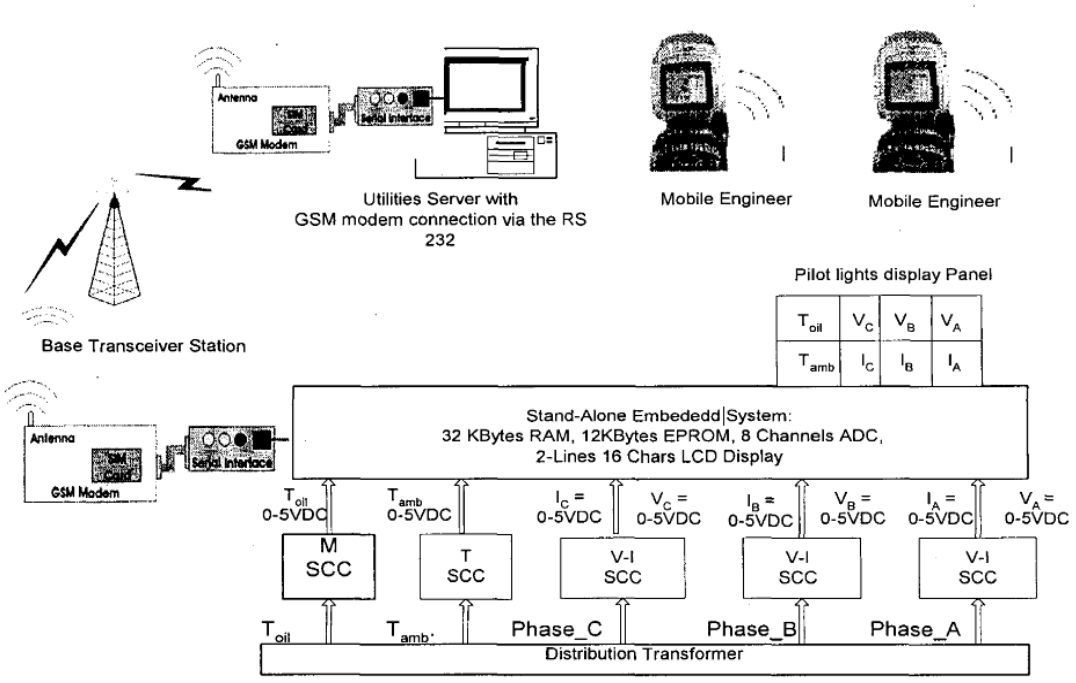


Figure 1. , System hardware architecture

The above scheme depicts the sequence of methodologies followed in the monitoring of distribution transformer via GSM technology[8].

- First sensors which are installed at the transformer site sense the various parameters of transformers and convert into analog signal to be processed in signal conditioning circuits
- Next the SCC consisting of opamps and resistors manipulates the analog signal to a compatible value so that can be read by the embedded system.

- Next the signal is passed through microcontroller. The ADC is used to read the parameters, built-in EEPROM is used to host the embedded software algorithm that takes care of the parameters acquisition, processing, displaying, transmitting and receiving. The built-in EEPROM is used to save the online measured parameters along with their hourly and daily averages.
- The GSM modem is interfaced with the microcontroller through RS 232 adapter by which it upload and download SMS messages that contain information related to the transformer parameters and status.
- This GSM modem then sends this SMS to mobile users containing information about parameters value of the distribution transformers.

3.2 SENSORS

Sensors are installed on transformer site which reads and measures the physical quantity from the distribution transformer and then it converts it into the analog signal. Sensor are used for sensing load current, ambient temperature, winding temperature, oil temperature and oil level. A sensor is a device which receives and responds to a signal when touched. A multitude of different measurable variables can be collected for on-line monitoring[1]. However, it is very rarely useful to use the entire spectrum. Therefore, sensor technology must be adjusted to the specific requirements of a particular transformer depending on their age and condition.

Following general set-up of sensors for example is proposed[1]-[3] for the use at a Distribution transformer:

• PT100 to measure top oil temperature
• PT100 to measure ambient temperature
• C.T to measure load current (single phase)
• Determination of voltage at measurement tap of bushing (three phase)

• Estimation of oil pressure of bushing

• Sensor to measure humidity in oil

• Sensor for measuring gas-in-oil content

It is fundamental to measure the electrical variables load current and operating voltage directly at the transformer. A bushing-type current transformers is used for load current measurement.

For the gas-in-oil detection a Hydran[3] sensor is used which reads a composite value of gases in ppm (H2 (100%), CO (18%), C2H2(8%), C2H4 (1,5%)). As hydrogen is a key gas for problems in the active part, an increase in the output signal of the sensor is an indication for irregularities such as partial discharge or hot spots. The evaluation of this measuring signal, together with the dependency on the temperature of the oil and the load current, provides a reliable basis for the continuous operation of the transformer. In the event of an increase of gas-in-oil content, an immediate reaction can be effected via an off-line dissolved gas analysis to determine the concentration of the other components dissolved in the oil in order to clarify the cause of the potential damage.

A capacitive thin film[1] sensor is used for the detection of moisture in oil. There are several causes for an increase of water-in-oil content. After improper shipping and erection of the transformer on-site the oil can be contaminated with water. Breathing of the transformer can cause absorption of moisture by the oil in the conservator. Due to the fact that water is a result and also an origin of paper degradation the water-in-oil content is an important indicator for the condition of winding insulation.

The voltage applied to the transformer is acquired at the measuring tap of the capacitor bushing by means of a voltage sensor. It acts with the capacity of the bushing as a voltage divider. This enables not only the measurement of the operational voltage but also the detection of overvoltages, because due to its design the voltage sensor has a bandwidth up to some MHz. The output of the voltage sensor is connected to a peak sampler to detect the amplitude of overvoltages by the monitoring system.

Sl No	Parameter	Sensors Used
1	Phase Current	Current Transformer
2	Phase Voltage	Voltage Transformer
3	Oil temperature	Thermistor, TPT-32
4	Oil Level	R-Series,Rxl oil level sensor

5	Winding Temperature	RTD Sensors
6	Gas Content in Oil	Hydran Sensor

Table 1:Sensors for different transformer parameters

3.3 CIRCUIT DIGRAM OF THE MODEL

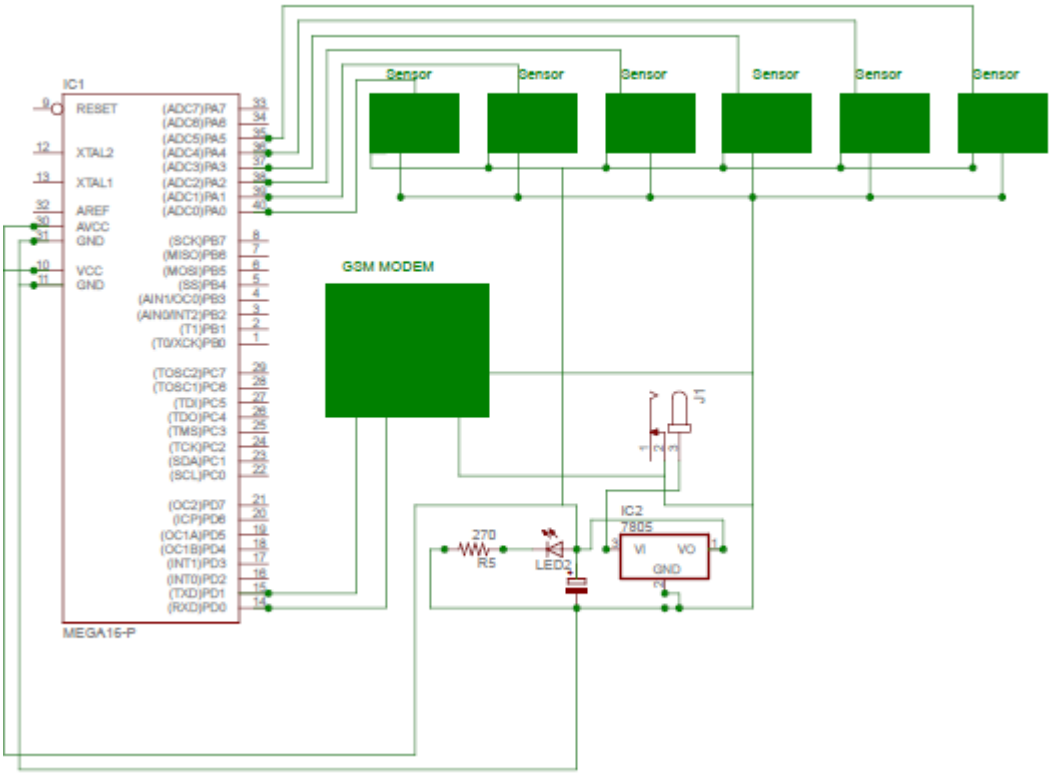
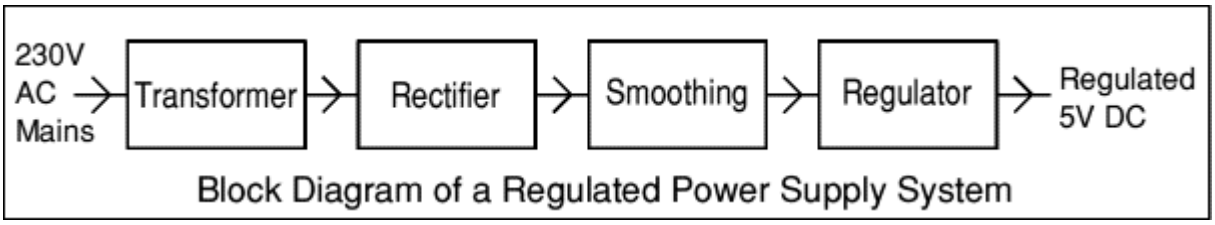


Figure 2: Circuit diagram of the Model

3.4 POWER SUPPLY

Power supply is the circuit from which we get a desired dc voltage to run the other circuits. The voltage we get from the main line is 230V AC but the other components of our circuit require 5V DC. Hence a step-down transformer is used to get 12V AC which is later converted to 12V DC using a rectifier. The output of rectifier still contains some ripples even though it is a DC signal due to which it is called as Pulsating DC. To remove the ripples and obtain smoothed DC power filter circuits are used. Here a capacitor is used. The 12V DC is rated down to 5V using a positive voltage regulator chip 7805. Thus a fixed DC voltage of 5V is obtained.

A 5V regulated supply is taken as followed:



Block Diagram of a Regulated Power Supply System

Each of the blocks is described below:

- Transformer - steps down high voltage AC mains to low voltage AC.
- Rectifier - converts AC to DC, but the DC output is varying.
- Smoothing - smoothes the DC from varying greatly to a small ripple.
- Regulator - eliminates ripple by setting DC output to a fixed voltage.

3.4.1 TRANSFORMER

Transformer is the electrical device that converts one voltage to another with little loss of power. Transformers work only with AC. There are two types of transformers as Step-up and Step-down transformer. Step-up transformers steps up voltage, step-down transformers steps down voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage to a safer low voltage. Here a step down transformer is used to get 12V AC from the supply i.e. 230V AC.

3.4.2 RECTIFIERS

A rectifier is a circuit that converts AC signals to DC. A rectifier circuit is made using diodes. There are two types of rectifier circuits as Half-wave rectifier and Full-wave rectifier depending upon the DC signal generated.

3.4.3 SMOOTHING

Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as reservoir, supplying current to the output when the varying DC voltage from the rectifier is decreasing. The diagram shows the unsmoothed varying DC and the smoothed DC. The capacitor charges quickly to the peak of the varying DC and then discharges as it supplies current to the output. Here the capacitor of 330uF is used as a smoothing circuit.

3.4.4 VOLTAGE REGULATION

Voltage regulators produce fixed DC output voltage from variable DC (a small amount of AC on it). Fixed output is obtained by connecting the voltage regulator at the output of the filtered DC. It can also be used in circuits to get low DC voltage from high DC voltage (for example we use 7805 to get 5V from 12V). Two types of voltage regulators are

1. fixed voltage regulators (78xx, 79xx)
2. Variable voltage regulators (LM317)

3.5 MICROCONTROLLER

Microcontroller is defined as a system on computer chip which includes number of peripherals like RAM, EEPROM, etc. required to perform some predefined task. There are number of popular families of microcontrollers which are used in different applications as per their capability and feasibility to perform various task, mostly used of these are 8051, AVR and PIC microcontrollers. In this subject we will introduce you with AVR family of microcontrollers. AVR is an 8-bit microcontroller belonging to the family of Reduced Instruction Set Computer (RISC). In RISC architecture the instruction set of the computer are not only fewer in number but also simpler and faster in operation. The other type is CISC . We will explore more on this when we will learn about the architecture of AVR microcontrollers in following section.

The microcontroller transmits and receives 8-bit data. The input/output registers available are also of 8-bits. The AVR families controllers have register based architecture which means that both the operands for an operation are stored in a register and the result of the operation is also stored in a register.

Discussing about AVR we will be talking on Atmega16 microcontroller, which is 40-pin IC and it belong to mega AVR category of AVR family. Some of the key features of Atmega16 are:

- 16KB Flash memory
- 1KB SRAM
- 512 Bytes EEPROM
- 40-Pin DIP
- 8-Channel 10 bit ADC
- Two 8 bit Timers/Counters
- One 16 bit Timer/Counter
- 4 PWM Channels
- In System Programmer (ISP)
- Serial USART
- SPI Interface
- Digital to Analog Comparator.

3.5.1 ARCHITECTURE OF AVR

The AVR microcontrollers are[5] based on advanced RISC architecture and it consist 32 x 8-bit general purpose working registers. Within one single clock cycle, AVR will take inputs from two general purpose registers and put them to ALU to carry out the operation, and will transfer back the result to any arbitrary register. The ALU performs arithmetic as well as

logical operations over the inputs from the register or between the register. We can see that AVR does not have any register like accumulator like in 8051 family of microcontrollers; the operations can be performed between any registers and can be stored in any register.

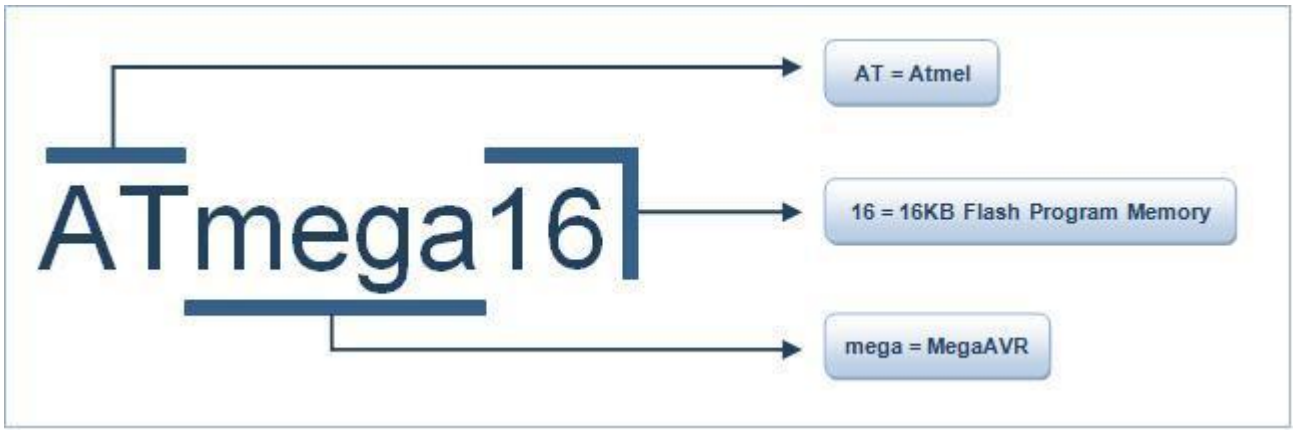
AVR follows Harvard Architecture format in which the processor which is equipped with the separate memories [6]and buses for Program and the Data information. Here when an instruction is executed, the next instruction will be pre-fetched from the program memory. Since AVR performs a single cycle execution, it means that AVR can execute 1 million instructions per second if the cycle frequency is 1MHz. If the operating frequency of the microcontroller is higher, then processing speed is also higher. We should optimize the power consumption with processing speed and hence should select the operating frequency accordingly.

Two types of Atmega16 microcontroller are:

1. Atmega16:- Operating frequency range is 0 – 16 MHz
2. Atmega16L:- Operating frequency range is 0 – 8 MHz

If a crystal of 8 MHz = 8 x 10^6 Hertz = 8 Million cyclesis used, then AVR can execute 8 million instructions.

The AT refers to manufacturer Atmel, Mega means the microcontroller belong to Mega AVR category, 16 gives the memory of the controller, which is 16KB.



Architecture Diagram: [6]Atmega16

Described below explains the building blocks of Atmega16 architecture:

· I/O Ports: Atmega16 has four (PORTA, PORTB, PORTC and PORTD) 8-bit input-output ports.

· Internal Calibrated Oscillator: Atmega16 has an internal oscillator for driving its clock. By default Atmega16 will operate at internal calibrated oscillator of 1 MHz The maximum frequency of internal oscillator will be 8Mhz. In other way ATmega16 can be operated using an external crystal oscillator with having a maximum frequency of 16MHz.

. ADC Interface: Atmega16 has a 8 channel ADC (Analog to Digital Converter) and a resolution of 10-bits. ADC reads the analog input for e.g., a sensor input and converting it into digital information which the microcontroller understands.

·Timers/Counters: Atmega16 consists of two 8-bit and one 16-bit timer/counter. Timers are useful for generating precise actions for e.g., creating time delays among two operations.

·Watchdog Timer: Watchdog timer is present alongwith internal oscillator. Watchdog timer monitors continuosly and resets the controller if the code is stuck at any execution action for more than a fixed time interval.

· Interrupts: Atmega16 consists of 21 interrupts sources out of which four are external. The rest are internal interrupts which is supported by the peripherals like USART, ADC, and Timers etc.

.USART: Universal Synchronous and Asynchronous Receiver and Transmitter interface is available to be interfaced with external device capable of communicating serially (data transmission bit by bit).

·General Purpose Registers: Atmega16 has 32 general purpose registers which are coupled directly with the Arithmetic Logical Unit (ALU) of CPU.

· ISP: AVR family have In System Programmable Flash Memory which is programmed without removing the IC from the circuit, ISP allows to reprogram the controller while it is inside the application circuit.

.DAC: Atmega16 is also equipped with a Digital to Analog Converter (DAC) interface which can be used for reverse action. DAC can be used when there is a need of converting a digital signal to analog signal.

. Memory: Atmega16 consist of three different memory sections:

1. Flash EEPROM: Flash EEPROM or simple flash memory is used to store the program burnt by the user on to the microcontroller. It is easily erasable electrically as a single unit. Flash memory is non-volatile i.e. the programme is retained even if the power is removed. Atmega16 is available with 16KB of in system programmable Flash EEPROM.

2. Byte Addressable EEPROM: This is also a nonvolatile memory used to store data of certain variables. Atmega16 has 512 bytes of EEPROM; this memory can be useful for storing the lock.

3. SRAM: Static Random Access Memory is the volatile memory of microcontroller i.e., data is lost as soon as power is removed. Atmega16 is equipped with 1KB of internal SRAM. A small portion of SRAM for general purpose registers used by CPU and some for the peripheral subsystems of the microcontroller.

· SPI: Serial Peripheral Interface, SPI port is used in serial communication between two devices with a common clock source. The data transmitting rate of SPI is more than USART.

· TWI: Two Wire Interface (TWI) can be used to set up a network of devices, many devices can be connected over TWI interface forming a network, the devices simultaneously transmit and receive and having own unique address.

PIN DIAGRAM :

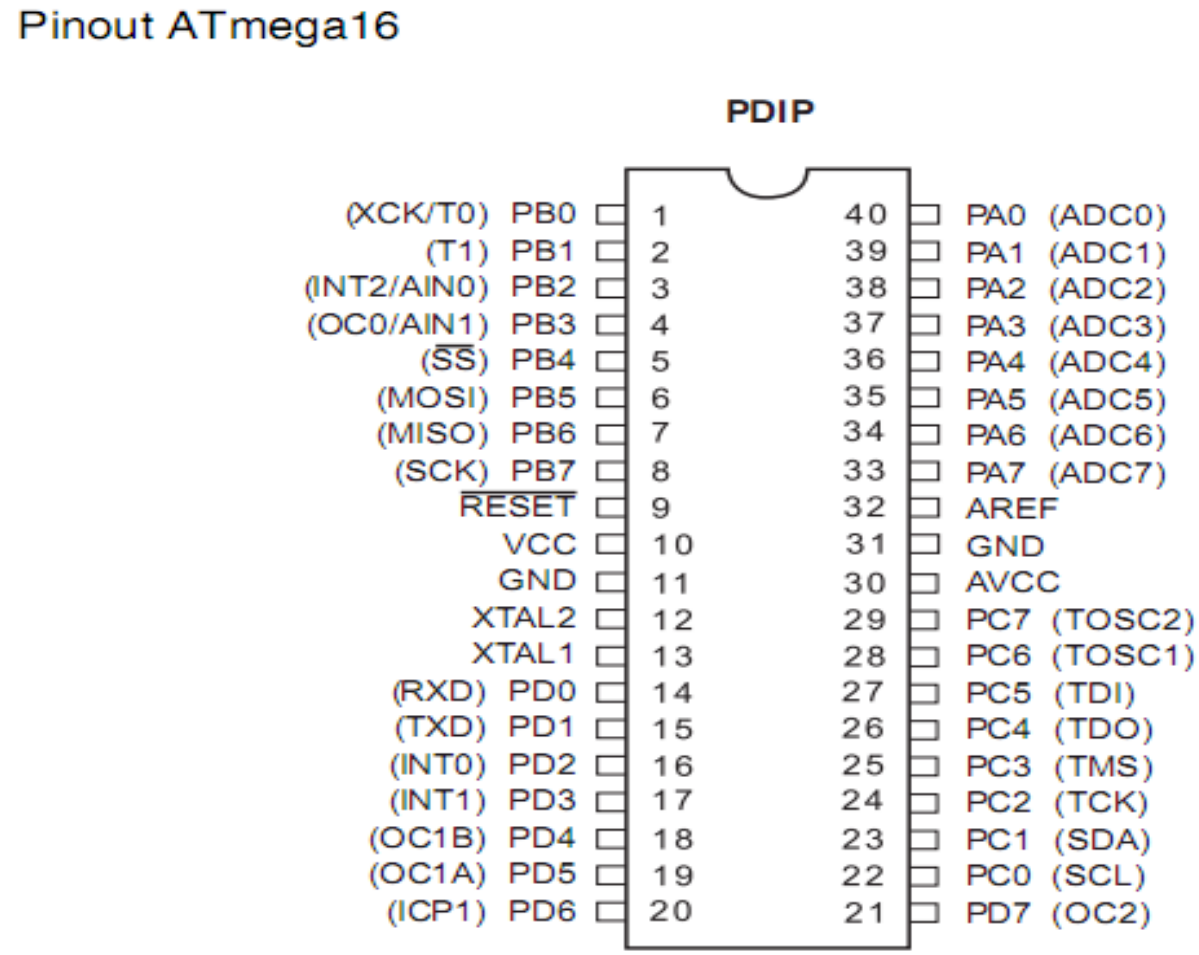


Figure 3:Pin Diagram of ATmega16[5]

3.5.2 PIN DESCRIPTIONS

VCC Digital supply voltage.

GND Ground.

Port A (PA7..PA0) Port A serves the analog inputs to the A/D Converter.

Port A is an 8-bit bi-directional I/O port, if A/D Converter is not used. The Port A output buffers have symmetrical[6] drive characteristics having high sink and source capability. When the pins PA0 to PA7 are used as the inputs and are pulled low externally, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when the condition is reset, even if the clock is not running.

Port B (PB7..PB0) Port B is an 8-bit bi-directional I/O port having internal pull-up resistors (selected for each bit). The Port B output buffers symmetrically drive characteristics having both high sink and source capability. As the inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when the condition is reset, even if the clock is not running.

Port C (PC7..PC0) Port C is an 8-bit bi-directional I/O port having internal pull-up resistors (selected for each bit). The Port C output buffers symmetrically drive characteristics having both high sink and source capability. Port C pins as inputs that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a condition is reset, even if the clock is not running.When the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs. The TD0 pin is tri-stated untill TAP states that the data shifted out are entered.

Port D (PD7..PD0) Port D is an 8-bit bi-directional I/O port having internal pull-up resistors (selected for each bit). The Port D output buffers symmetrically drive characteristics having both high sink and source capability. As inputs, Port D[6] pins that are externally pulled low will source the current if the pull-up resistors are activated. The Port D pins are tri-stated when a condition is reset, even if the clock is not running.

RESET Reset Input. A low level on this pin for longer period than the minimum pulse length will give a reset, even if the clock is not running. Shorter pulses will not generate a reset.

XTAL1 Is an input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2 Is an output from the inverting Oscillator amplifier.

AVCC AVCC is the supply voltage pin for Port A and the A/D Converter. It is generally externally connected to VCC, if the ADC is not used. If the ADC is used, it is connected to VCC through a low-pass filter.

AREF Is the analog reference pin for the A/D Converter.

I/O PORTS

At mega 16 have 32 general purpose digital I/O pins. To every pin, there are 3 bits in 3 different registers which control its function. Let's say we are talking about the pin PA0. The three registers involved for this pin are DDRA, PORTA and PINA, similarly the corresponding bits are DDRA0, PORTA0 and PINA0.

DDR - is the Data Direction Register – 1 is written to DDRA0 making the pin PA0 act like an output pin and writing 0 makes it an input pin.

Code example:

DDRC=130;

or,

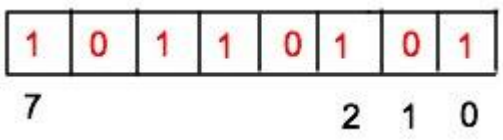
DDRC= 0b 10000010;

or,

DDRC= 0x 82;

Both the above statements will make the PC1 and PC7 as output and rest as input.

It is to be noted that writing some value onto a register simply means that the bits of the register will attain values such that the binary number represented by all the 8 bits of the register together equals the number assigned to them. e.g. writing 0b10110101 means the bits in the register will become like this:



PORT register - If DDRA0 is set as 1,

- If 1 is written to PORTA0 gives a high output on pin PA0
- If 0 is written to PORTA0 gives a low output on pin PA0

If DDRA0 is set to 0 (input),

- If 1 is written to PORTA0 simply pulls up the pin to Vcc via 100k resistance
- If 0 is written to PORTA0 makes the pin tri-stated . This means that in the absence of input from outside the pin will just have some random value.

PIN register - This register is used to read the digital value of the pin. It can be thought of as actually connected to MCU physical pins. If voltage of the pin (either in case of input or output) at any instant is low it will read as 0 otherwise 1.
For example

Int read;

read=PINB; // stores the value of 8 bit PINB register in the variable read

or

read=PINB & 4; // This statement stores the value of PB2 bit in read.

3.6 LCD (Liquid Crystal Display)

The display used is 16x2 LCD (Liquid Crystal Display); which means 16 characters per line by 2 lines.The standard is referred as HD44780U, which refers to the controller chip which receives data from an external source (Here Atmega16) and communicates directly with the LCD. Here 8-bit mode of LCD is used, i.e., using 8-bit data bus.

The three control lines are **EN**, **RS**, and **RW**.

The **EN** line is called "Enable." This[5] control line is used for telling the LCD that we are sending data. For sending data to the LCD, the program should make sure that the line is low (0) and then set the other two control lines or put data on the data bus. When the other lines

are ready completely, bring **EN** high (1) and should wait for the minimum time required by the LCD datasheet and end by bringing it low (0) again.

The **RS** line is "Register Select" line. When RS is low (0), the data is treated as a command or special instruction (such as clear screen, position cursor, etc.). When the RS is high (1), the data sent is text data which is displayed on the screen. For example, to display the letter "B" on the screen you would set RS high.

The **RW** line is "Read/Write" control line. When RW is low (0), the information on the data bus is written to the LCD. When RW is high (1), the program is effectively questioning (or reading) the LCD. Only one instruction ("Get LCD status") is read command. All the others are write commands--so RW will always be low.

In case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

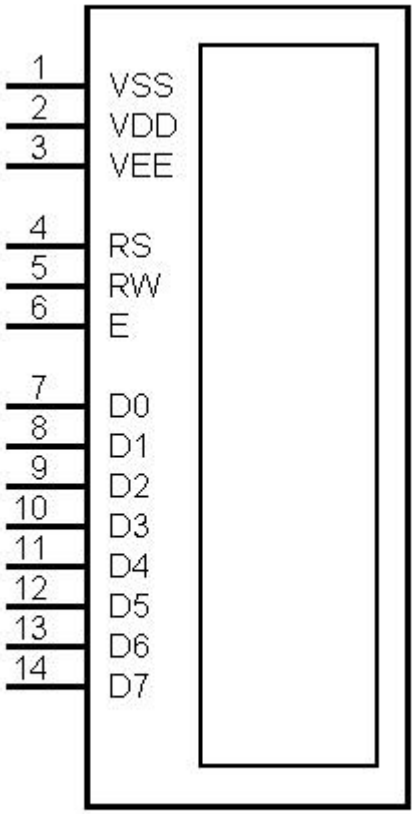


Figure 4: Pin Diagram of LCD[6]

Pin no	Name	Description
1	VSS	GND
2	VCC	Power supply (+5v)
3	VEE	Contrast Adjust
4	RS	0=Instruction Input 1=Data Input
5	R/W	0=Write to LCD 1=Read From LCD
6	EN	Enable Signal
7	D0	Bit 0 LSB

8	D1	Bit 1
9	D2	Bit 2
10	D3	Bit 3
11	D4	Bit 4
12	D5	Bit 5
13	D6	Bit 6
14	D7	Bit 7 MSB

Table 2: Pin Description of LCD[6]

3.7 GSM MODEM

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem is like a dial-up modem. The basic difference between them is the dial-up modem sends and receives data through a fixed telephone line while the wireless modem sends and receives data through waves. Like a GSM mobile phone, a GSM[7] modem also requires a SIM card from a wireless carrier to operate.

Figure 5: GSM Modem

SIM 300 is a Fixed Cellular Terminal (FCT) used for data applications. It is a compact and portable terminal which satisfy various data[7] communication over GSM. It also can be connected to a computer with a standard RS232C serial port. SIM 300 offers features like Short Message Services (SMS), Data Services (sending and receiving data files), Fax

Services and data file connectivity through wire is not available or not possible. The SIM 300 is very easy to set up. It also finds its applications in IT companies, Banks,Financial Institutions, Service Providers, Far away Project Sites, and other business establishments.

Product concept:

SIM300 designed for global market is a Tri-band GSM/GPRS engine which works on the frequencies 900MHz EGSM, 1800MHz DCS and 1900MHz PCS. SIM300 features GPRS multi slot class10/ class8 and supports the GPRS coding schemes CS-i, CS-ii,CS-iii and CS-iv.

SIM card interface:

We can use AT Command for getting information in SIM card. The SIM interface supports the operation of the GSM Phase 1 specification and also supports the operation of the new GSM Phase 2 and specification for FAST 64kbps SIM (intended to use having a SIM application Tool-kit). Both the 1.8V and 3.0V SIM Cards are supported. The SIM interface get its power from an internal regulator in the module having nominal voltage 2.8V. All the pins are reset as outputs driving low.

Operation: AT commands are used by the computers to control modems. Both the GSM modems and dial-up modems support a fixed set of standard AT commands. GSM modem can be used like a dial-up modem. Apart from the standard AT commands, GSM modems also support an extended set of AT commands. These extended set of AT commands are defined in the GSM standards. With the extended AT commands, several things are done:

- To read,write and delete SMS messages.
- To send SMS messages.
- To monitor the signal strength.
- To monitor the charging status and charge level of the battery.
- Reading, writing and searching phone book entries.

The number of SMS messages processed by a GSM modem per minute is very low -- only six to ten SMS messages per minute.

Network status indication LED lamp

State SIM300 function

Off- SIM300 is not running

64ms On/ 0.8 sec Off- SIM300 does not find the network

64ms On/ 3Sec off- SIM300 find the network

64 ms on / 0.3sec Off- GPRS communication

3.7.1 INSTRUCTION OF GSM MODEM

AT commands: AT commands are the instructions used for controlling a modem. AT stands for Attention. Each and every command line starts with "AT" or "at". Beacause of this modem commands are called AT commands. Many of the commands are also used for controlling wired dial-up modems. These are supported by GSM/GPRS modems and mobile phones. Apart from this common AT command set, GSM/GPRS modems and mobile phones also support an AT command sets which are specific to the GSM technology, which also includes SMS-related commands.

Basic Commands and Extended Commands:

There are two types of AT commands: They are basic commands and extended commands.

- Basic commands are AT commands that do not start with "+". For example, D (Dial), A (Answer), H (Hook control) and O (Return to online data state) are basic commands.
- Extended commands are AT commands that start with "+". All GSM AT commands are extended commands. For example, +CMGS (Send SMS message), +CMSS (Send SMS message from storage), +CMGL (List SMS messages) and +CMGR (Read SMS messages) are extended commands.

3.7.2 SERIAL COMMUNICATION

In our model serial communication from modem to[5] microcontroller are done by connecting Txd and Rxd pins to modem Rxd and Txd pin respectively. And the third pin of modem is grounded. In our hardware architecture we have interface modem to microcontroller directly without the use of Max232 or RS232 and having proper results with proper communication. Max232 or RS232 both are used as logic converter. They both can work in CMOS logic level or TTL logic level. If microcontroller works in TTL level and GSM Modem works in CMOS level then logic converter like RS232 is interfaced to bring same logic level. But in our model both Microcontroller and GSM Modem works in TTL logic level so here we have not used Max232 0r RS232.Direct connections between modem and microcontroller is done.

CHAPTER 4

SOFTWARE IMPLEMENTATION

This chapter describes about the software implementation of the project.This discuss about the programming and the software tools used and how output is obtained by programming.

4.1 SOFTWARE TOOLS

AVR Studio and Pro Load are two softwares used to program microcontroller.

4.2 PROGRAMMING MICROCONTROLLER

The compiler for high level language helps to reduce production time.For programming ATmega16L AVR Studio is used.The programming is done in embedded C language. The

compilation of the C program converts it into machine language file (.hex). This is the only language the microcontroller will understand, because it has the original program code converted into a hexadecimal format.During this process some errors and warnings occurs. If there are no errors and warnings then run the program, the system performs all the given tasks and behaves as expected the software developed. If not the whole procedure is repeated again.

4.2.1 AVR STUDIO

AVR studio is software used where machine language code is written and compiled. After compilation machine source code is[6] converted to hex code to be burnt into the microcontroller. The program is written in C language code.

4.2.2 PROLOAD

Proload is software that accepts only hex files. After the machine code is converted into hex code, that hex code has to be burnt into the microcontroller which done by the Proload. Proload is a programmer which contains a microcontroller in it other than the one which is to be programmed. The program is written in the Proload microcontroller in such a way that it accepts the hex file from the AVR Studio and burns this hex file into the microcontroller which is to be programmed. The Proload programmer kit requires power supply to operate, this power supply is given by the power supply circuit. It is noted that this programmer kit contains a power supply section in the board but in order to switch on the power supply, a source is required. This is accomplished from the power supply board with an output of 12 V.

- Microcontroller Software Compiler generates a Hex file.
- Hex file accepted and sent to MCU program Loader.
- Hex file programmed into Target Microcontroller device.

4.3 PROJECT SOURCE CODE

Main Code

```
#include<avr/io.h>                //including input output header file
#define F_CPU 8000000
#include<avr/delay.h>
#define LCD_DATA_PORT  PORTC
```

```
#include<avr/lcd.h>
#include<avr/adc.h>
#include<avr/usart.h>
#define REF 500
unsigned char gsm_sim_status(void);
void gsm_set_format(void);
void send_message(unsigned char *number, unsigned char *message);
unsigned char temp=0,msg_send=1;
unsigned char gsm_buffer[300],gsm_buffer1[300],gsm_buffer2[20];
unsigned char sim1[]="+cpin: ready",sim2[]="+cpin: not inserted", sim3[]="+CMS ERROR:
515";
unsigned char student_numbers[11]="9439197325";
unsigned char message0[]={"PH1 Over Current"};
unsigned char message1[]={"PH1 Over Voltage"};
unsigned char message2[]={"PH2 Over Current"};
unsigned char message3[]={"PH2 Over Voltage"};
unsigned char message4[]={"PH3 Over Current"};
unsigned char message5[]={"PH3 Over Voltage"};
void main()
{
        DDRC=0xff;   //lcd data port declairing as output port
        DDRB=0x07;  //lcd command port
        DDRA=0b10000000;//PA7-sim status
        DDRD=0b11111100;// for signifing Txd and Rxd
       int sensor0_output=0,sensor1_output=0,sensor2_output=0,
            sensor3_output=0,sensor4_output=0,sensor5_output=0;
        adc_init();
        usart_init();
        lcd_init();
        lcd_string_write("WELCOME");
```

```
if(gsm_sim_status())
{
        lcd_command_write(0x01);            //clear LCD
        lcd_string_write("SIM Ready");
        PORTA|=_BV(PA7);                    //Setting PA7 bit
}
else
{
        lcd_command_write(0x01);
        lcd_string_write("SIM NOT Inserted");
        PORTA&=~(_BV(PA7));                 //Resetting PA7 bit
}
 _delay_ms(500);
lcd_command_write(0xc0);
lcd_string_write("wait a while");
//delay not to get cms error 515
for(unsigned char i=0;i<80;i++)
        _delay_ms(500);
lcd_command_write(0x01);                    //LCD CLear
lcd_string_write("proceeding now..");
while(1)
{
        sensor0_output=read_adc_channel(0);
        sensor1_output=read_adc_channel(1);
        sensor2_output=read_adc_channel(2);
        sensor3_output=read_adc_channel(3);
        sensor4_output=read_adc_channel(4);
        sensor5_output=read_adc_channel(5);
         if(msg_send==1)
```

```
{
        lcd_command_write(0xc0);
        //sensor 0
        if(sensor0_output > REF)
        {
                lcd_string_write("1 ");
                gsm_send_message(student_numbers,message0);
                msg_send=0;
                PORTD|=_BV(PD2);
        }
        else
        {
                PORTD&=~(_BV(PD2));
        }
         //sensor1
        if(sensor1_output > REF)
        {
                lcd_string_write("2 ");
                gsm_send_message(student_numbers,message1);
                msg_send=0;
                PORTD|=_BV(PD3);
        }
        else
        {
                PORTD&=~(_BV(PD3));
        }
        //sensor2
        if(sensor2_output > REF)
        {
```

```
        lcd_string_write("3 ");
        gsm_send_message(student_numbers,message2);
        msg_send=0;
        PORTD|=_BV(PD4);
}
else
{
        PORTD&=~(_BV(PD4));
}
//sensor3
if(sensor3_output > REF)
{
        lcd_string_write("4 ");
        gsm_send_message(student_numbers,message3);
        msg_send=0;
        PORTD|=_BV(PD5);
}
else
{
        PORTD&=~(_BV(PD5));
}
//sensor4
if(sensor4_output > REF)
{
        lcd_string_write("5 ");
        gsm_send_message(student_numbers,message4);
        msg_send=0;
        PORTD|=_BV(PD6);
}
```

```
                else
                {
                        PORTD&=~(_BV(PD6));
                }
                  //sensor5
                if(sensor5_output > REF)
                {
                        lcd_string_write("6 ");
                        gsm_send_message(student_numbers,message5);
                        msg_send=0;
                        PORTD|=_BV(PD7);
                }
                else
                {
                        PORTD&=~(_BV(PD7));
                }
        }
    }
}
unsigned char gsm_sim_status(void)
{
    //rdy
    temp=usart_data_receive();
    temp=usart_data_receive();
    usart_string_receive(gsm_buffer,13);
    temp=usart_data_receive(); //cpin or cfun
    temp=usart_data_receive();
    temp=usart_data_receive();
    usart_string_receive(gsm_buffer,13);
```

```
temp=usart_data_receive();
if(!(strcasecmp(gsm_buffer,sim2)))
{
        //cpin+cfun
        temp=usart_data_receive();
        temp=usart_data_receive();
        usart_string_receive(gsm_buffer,13);
        temp=usart_data_receive();
        return 0;
}
else
{
        //cfun+cpin
        temp=usart_data_receive();
        temp=usart_data_receive();
        usart_string_receive(gsm_buffer,13);
        temp=usart_data_receive();
}
if(!(strcasecmp(gsm_buffer,sim1)))
{
        //call ready
        temp=usart_data_receive();
        temp=usart_data_receive();
        usart_string_receive(gsm_buffer,13);
        temp=usart_data_receive();
        return 1;
}
Else
  ;
```

```
}
void gsm_set_format(void)
{
        usart_data_transmit('a');
        temp=usart_data_receive();
        usart_data_transmit('t');
        temp=usart_data_receive();
        usart_data_transmit('+');
        temp=usart_data_receive();
        usart_data_transmit('c');
        temp=usart_data_receive();
        usart_data_transmit('m');
        temp=usart_data_receive();
        usart_data_transmit('g');
        temp=usart_data_receive();
        usart_data_transmit('f');
        temp=usart_data_receive();
        usart_data_transmit('=');
        temp=usart_data_receive();
        usart_data_transmit('1');
        temp=usart_data_receive();
        usart_data_transmit(13);
        temp=usart_data_receive();
        temp=usart_data_receive();
        temp=usart_data_receive();
        usart_string_receive(gsm_buffer,13);
        temp=usart_data_receive();
        usart_data_transmit('a');
        temp=usart_data_receive();
```

```
	usart_data_transmit('t');
	temp=usart_data_receive();
	usart_data_transmit('&');
	temp=usart_data_receive();
	usart_data_transmit('w');
	temp=usart_data_receive();
	usart_data_transmit(13);
	temp=usart_data_receive();
	temp=usart_data_receive();
	temp=usart_data_receive();
	usart_string_receive(gsm_buffer,13);
	temp=usart_data_receive();
}
void gsm_send_message(unsigned char *number, unsigned char *message)
{
	//lcd_command_write(0xc0);
	//lcd_data_write('a');
	usart_data_transmit('a');//lcd_data_write('a');
	temp=usart_data_receive();//lcd_data_write(temp);
	usart_data_transmit('t');
	temp=usart_data_receive();//lcd_data_write(temp);
	usart_data_transmit('+');
	temp=usart_data_receive();//lcd_data_write(temp);
	usart_data_transmit('c');
	temp=usart_data_receive(); //lcd_data_write(temp);
	usart_data_transmit('m');
	temp=usart_data_receive(); //lcd_data_write(temp);
	usart_data_transmit('g');
	temp=usart_data_receive(); //lcd_data_write(temp);
```

```
usart_data_transmit('s');
temp=usart_data_receive();//lcd_data_write(temp);
usart_data_transmit('=');
temp=usart_data_receive();
usart_data_transmit("");
temp=usart_data_receive();
for(unsigned char i=0;i<10;i++)
{
        usart_data_transmit(number[i]);
        temp=usart_data_receive();
}
usart_data_transmit("");
temp=usart_data_receive();
usart_data_transmit(13);
temp=usart_data_receive();
temp=usart_data_receive();
temp=usart_data_receive();
temp=usart_data_receive();
_delay_ms(1);
for(unsigned char i=0;message[i]!='\0';i++)
{
        usart_data_transmit(message[i]);
        temp=usart_data_receive();
}
usart_data_transmit(0x1A);//ctrl+Z
temp=usart_data_receive();
temp=usart_data_receive();
temp=usart_data_receive();
usart_string_receive(gsm_buffer,13);
```

```
        temp=usart_data_receive();
         //lcd_command_write(0xc0);
        //lcd_string_write(gsm_buffer);
        //_delay_ms(500);
        if(!(strcasecmp(gsm_buffer,sim3)))
        {
                lcd_command_write(0x01);
                lcd_string_write("msg sending failed");
                //lcd_command_write(0xc0);
                //lcd_string_write("due to cms error");
        }
        else
        {
                temp=usart_data_receive();//lcd_data_write('1');
                temp=usart_data_receive();//lcd_data_write('2');
                usart_string_receive(gsm_buffer,13);
                temp=usart_data_receive();
                //lcd_data_write('3');
               //lcd_string_write(gsm_buffer);
        }
}
```

ADC Code

```
#ifndef _ADC_H_
#define _ADC_H_1
#include<avr/io.h>
#include<util/delay.h>
void adc_init(void);
int read_adc_channel(unsigned char channel);
void adc_init(void)
```

```
{
ADCSRA=(1<<ADEN)|(1<<ADSC)|(1<<ADATE)|(1<<ADPS2);
SFIOR=0x00;
}
int read_adc_channel(unsigned char channel)
{
        int adc_value;
        unsigned char temp;
         ADMUX=(1<<REFS0)|channel;
         _delay_ms(10);
        temp=ADCL;
        adc_value=ADCH;
        adc_value=(adc_value<<8)|temp;
         return adc_value;
}
 #endif
```

Serial Communication Code

```
#ifndef _USART_H_
#define_USART_H_ 1
#include<avr/io.h>
#include<util/delay.h>
void usart_init()
{
UBRRH = 0;
UBRRL =51;
UCSRB|= (1<<RXEN)|(1<<TXEN);
UCSRC |= (1 << URSEL)|(3<<UCSZ0);
}
void usart_data_transmit( unsigned char data )
```

```
{
while ( !( UCSRA & (1<<UDRE)) )
;
UDR = data;
}
unsigned char usart_data_receive( void )
{
while ( !(UCSRA & (1<<RXC)) )
;
return UDR;
}
unsigned char usart_string_transmit(unsigned char *string)
{
while(*string)
{
usart_data_transmit(*string++);
}
}
unsigned char *usart_string_receive(unsigned char *receive_string,unsigned char
terminating_character)
{
        unsigned char temp=0x00;
        for(unsigned char i=0;;i++)
        {
                *(receive_string+i)=usart_data_receive();
                if(*(receive_string+i)==terminating_character)
                        break;
                else
                temp++;
        }
```

```
        *(receive_string+temp)='\0';
        return receive_string;
}
#endif
```

LCD Code

```
#ifndef _LCD_H_
#define_LCD_H_1
#include<avr/io.h>
#include<util/delay.h>
#include<stdlib.h>
#ifndef LCD_DATA_PORT
#warning "LCD_DATA_PORT not defined for <avr/lcd.h.Default Port is PORTA>"
#define LCD_DATA_PORT  PORTA
#endif
#ifndef LCD_CONT_PORT
#warning "LCD_CONT_PORT not defined for <avr/lcd.h.Default Port is PORTB>"
#define LCD_CONT_PORT PORTB
#endif
#ifndef LCD_RS
#warning "LCD_RS not defined for <avr/lcd.h.Default Pin is PB0>"
#define LCD_RS PB0
#endif
#ifndef LCD_RW
#warning "LCD_RW not defined for <avr/lcd.h.Default Pin is PB1>"
#define LCD_RW PB1
#endif
#ifndef LCD_EN
#warning "LCD_EN not defined for <avr/lcd.h.Default Pin is PB2>"
```

```
#define LCD_EN PB2
#endif
void lcd_data_write(unsigned char data);
void lcd_command_write(unsigned char command);
void lcd_init();
void lcd_string_write(unsigned char *string);
void lcd_number_write(int number,unsigned char radix);
void lcd_data_write(unsigned char data)
{
LCD_CONT_PORT=_BV(LCD_EN)|_BV(LCD_RS);
LCD_DATA_PORT=data;
_delay_ms(1);
LCD_CONT_PORT=_BV(LCD_RS);
_delay_ms(1);
}
void lcd_command_write(unsigned char command)
{
LCD_CONT_PORT=_BV(LCD_EN);
LCD_DATA_PORT=command;
_delay_ms(1);
LCD_CONT_PORT=0x00;
_delay_ms(1);
}
void lcd_init()
{
        lcd_command_write(0x38);
        lcd_command_write(0x01);
        lcd_command_write(0x06);
        lcd_command_write(0x0e);
```

```
}
void lcd_string_write(unsigned char *string)
{
while (*string)
lcd_data_write(*string++);
}
void lcd_number_write(int number,unsigned char radix)
{
unsigned char *number_string="00000";
itoa(number,number_string,radix);
while (*number_string)
lcd_data_write(*number_string++);
}
#endif
```

CHAPTER 5

RESULTS

Nameplate rating of Distribution Transformer:

KVA rating: 500

Cooling type: oil natural

Voltage rating:

HV: 11KV

LV: 0.4KV

Current rating:

HV: 26.24A

LV: 666.70A

Impedance voltage: 3.92%

Vector Group: Dy11

Maximum Temperature Rise in oil: 45 deg.C

We need to monitor voltage and currents in phase.

Under normal condition

Ia=121.4A Vab=442.0V

Ib=108.8A Vbc=437.2V

Ic=138.5A Vca=440.3V

Taking the data of previous fault condition

Instantaneous load current in phases

Ia=318.6A

Ib=294.4A

Ic=333.1A

These currents of high magnitudes can't be fed directly to our designed system. It needs to be scaled down. For this purpose current transformer and potential transformer can be used in practice. The current and voltage after being scaled down is fed to microprocessor based system where it compare it with the reference value and take consequent action.

Regarding taking reference value, we have to take account the normal current, turn ratio of CT and PT, accuracy and associated errors. For example the

We are attempting to give that type situation using potentiometer. Using potentiometer we will vary the voltage and current in different phases.

The reference voltage is 2.44

By varying the potentiometer we change the voltage of phase 3 to 3.1

It is more than the reference voltage…

So the system will send message.

The following displays are obtained during the operation.

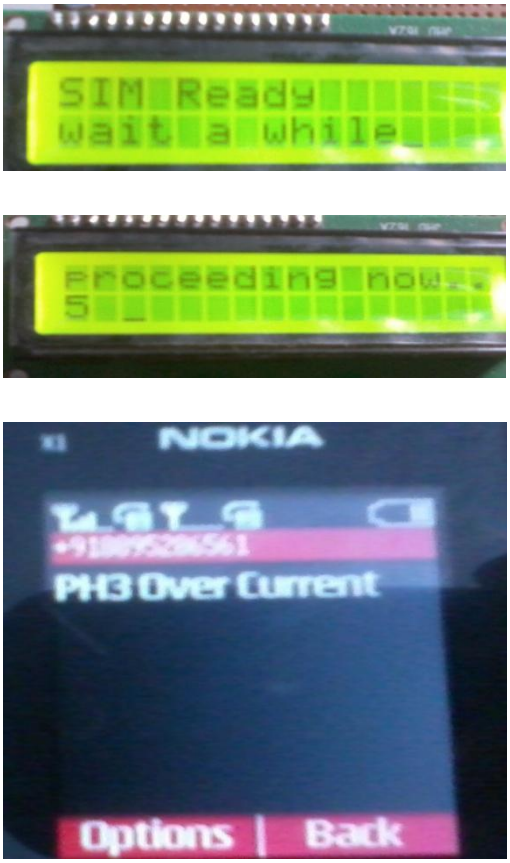


Figure 6: Results

CHAPTER 6

6.1 CONCLUSION

The gsm based monitoring of distribution transformer is quite useful as compared to manual monitring and also it is reliable as it is not possible to monitor always the oil level,oil temperature rise,ambient temperature rise,load current manually. After receiving of message of any abnormality we can take action immediately to prevent any catastrophic failures of distribution transformers. In a distribution network there are many distribution transformers and associating each transformer with such system, we can easily figure out that which

transformer is undergoing fault from the message sent to mobile. We need not have to check all transformers and corresponding phase currents and voltages and thus we can recover the system in less time. The time for receiving messages may vary due to the public GSM network traffic but still then it is effective than manual monitoring.

6.2 FUTURE WORK

A server module can be included to this system for receiving and storing transformer parameters information periodically about all the distribution transformers of a particular utility in a database application. This database will be a useful source of information on the utility transformers. Analysis of these stored data helps the utility in monitoring the operational behaviour of their distribution transformers and identify faults before any catastrophic failures thus resulting in significant cost saving as well as improving system reliability.

REFERENCES

[1] Leibfried, T, "Online monitors keep transformers in service", Computer Applications in Power, *IEEE, Volume:11 Issue: 3 , July 1998 Page(s):36 -42.*

[2] Chan, W. L, So, A.T.P. and Lai, L., L.; "Interment Based Transmission Substation Monitoring", *IEEE Transaction on Power Systems, Vol. 14, No. 1, February 1999, pp. 293-298.*

[3] Par S. Tenbohlen,T. Stirl, M. Rösner," Benefit of sensors for on-line monitoring systems for power transformers"

[4] T. D. Poyser, "An On-Line Microprocessor Based Transformer Analysis System to Improve the Availability and Uti'lization of Power Transformers". *IEEE Trans. On Power Apparatus and Systems, Volume PAS-102, April 1983, pp.957-962.*

[5] Muhammad Ali Mazidi , Janice Gillispie Mazidi, Rolin D.Mckinlay, The 8051 Microcontroller And Embedded Systems Using Assembly And C,Second Edition, Pearson Education, 2008, India.

[6] Microcontroller ATmega 16; www.atmel.com/Images/doc2466.pdf .

[7] Constantin Daniel Oancea," GSM Infrastructure Used for Data Transmission", 7th International Symposium on Advanced Topics in Electrical Engineering (ATEE), 2011 May 12-14, Page(s): 1 – 4.

[8] Abdul-Rahman AI-Ali, Abdul Khaliq & Muhammad Arshad," GSM-Based Distribution Transformer Monitoring System", *IEEE MELECON 2004, May 12-15,2004, Vol 3 Pages-999-1002, Croatia.*

APPENDIX

Power supply

Adapter	**230v ac/12v dc, wall wart type**
Voltage regulator	7805IC
Potentiometer	RJ50/RJR50
LED	$1.63 < \Delta V < 2.03$, ΔV =voltage drop, RED
Capacitor	Electrolytic,330micro farad

Embedded System

microcontroller	**Atmega 16L**
LCD	128*64 pixels

GSM modem

Gsm modem	**SIM 300**